ASSIGNMENT 2: GROUP 5

MACHINE LEARNING

Tutors: Dr Nguyen Hoang Tran, Iwan Budiman

Group Members:

Pujam Janghel (pjan2245),
Akanksha Patil (apat0532),
Mohammed Saif (moha6885

Contents

ABSTRACT

As we all know forest fires have devastating impact on the environment. In 2018, nearly 9 million acres were burned in the US alone. Uncontrolled fires often started accidentally by people, rampage and decimate forests (Hancock, n.d.). Once the forest fires gather momentum, it requires herculean efforts and resources to decelerate its progression. The data collected from these forest fires can be used to build machine learning algorithms that can predict the area of the forest fire from the given weather data.

Fast detection is imperative in controlling of this phenomenon. Weather data is recorded by meteorological stations and can easily be accessed. This weather data contains information such precipitation levels, wind speed and its direction, humidity and atmospheric pressure. The Dataset in our case contains data from the Montesinho natural park in northeastern region of Portugal. Support Vector Machines, Decision Trees and Random Forest algorithms were used to model the data. The performance of each were compared by the following metrics- Cross validation score, Root mean score error (RMSE) and Mean absolute deviation (MAD).

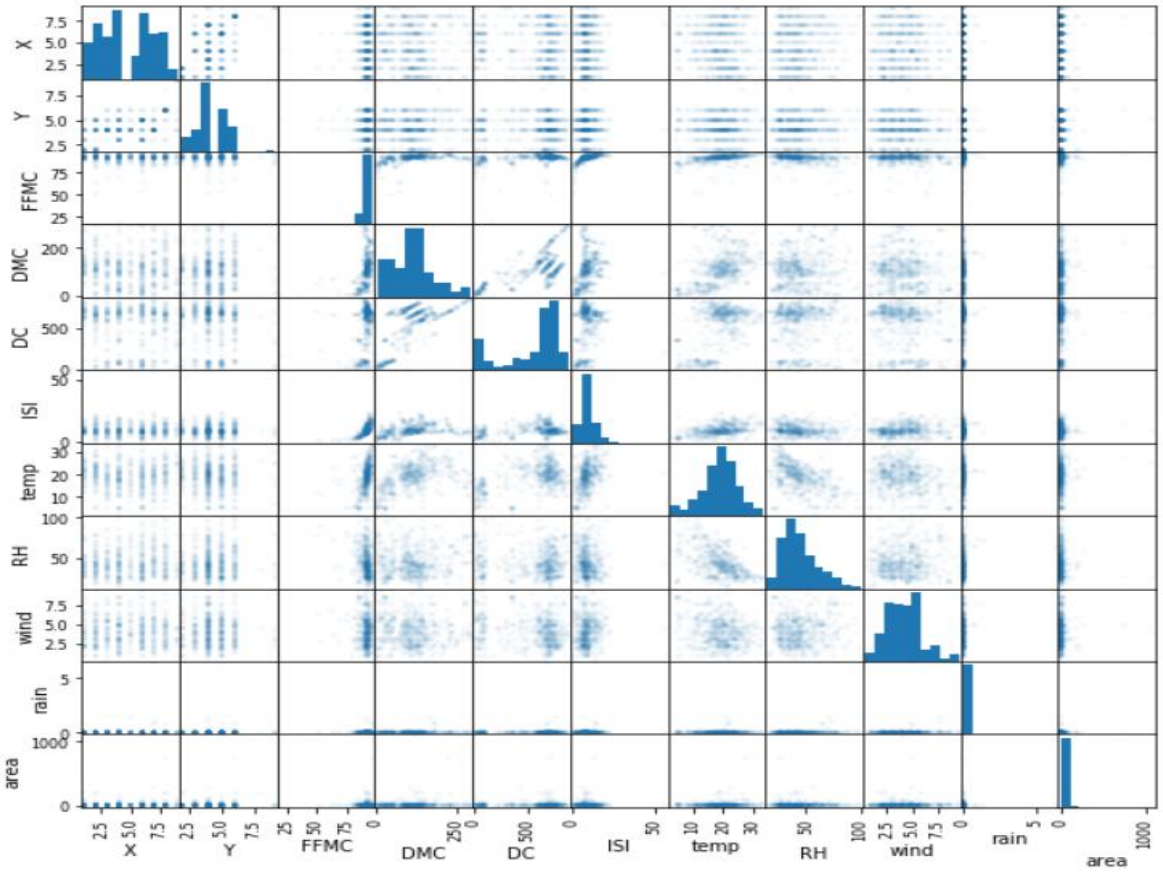
INTRODUCTION

Forest fires can cause significant damage if not controlled, and fire prevention is important to reduce the damage caused by them. Prevention can be achieved by modeling the relations between the fire threat (Forest fire area) and influence factors (features) (Daniela Stojanova, n.d.). Various data mining techniques can be applied on the dataset to build a model that can accurately predict the area of forest fire. The Montesinho natural park dataset used contains 517 records and has 13 attributes – Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Drought Code (DC), Initial Spread Index (ISI), temperature, relative humidity (RH), wind, rain, area, x-coordinate and y coordinate. It was recorded from January 2000 to December 2003. Every time there was a forest fires, details about the features mentioned above were noted and also the area burned by the forest fire. The area with 0 as an entry indicates forest fires where the less than $100m^2$ was burned (Paulo Cortez, n.d.). Non-Linear data does not follow a linear relationship and cannot be solved using simple regression. It needs complex data mining techniques for accurate prediction.

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.0
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.0
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.0
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.0

Data that follows linear pattern is relatively rare. Therefore, it is even more important to build algorithms than can work with the data. The algorithms we chose are Support Vector Machines, Decision Trees and Random Forest. Since this is a regression problem, the performance of each of the algorithms was measured using Root Mean Squared Error (RMSE) and Mean Absolute Deviation (MAD). The best model along with the best parameters were obtained using Grid Search Cross Validation (GridSearchCV) technique. In our case, SVR was our best regressor with RMSE value of 26.72.

Below the scatter matrix of the data.



PREVIOUS WORK: LITERATURE REVIEW

In the paper "A Data Mining Approach to Predict Forest Fires using Meteorological Data" published by Paulo Cortez and Anibal Morris, the dataset used was collected from Montesinho Natural Park in Portugal. The dataset consisted information about the weather and meteorological conditions present during the forest fires as well as the burned area. 5 Data mining techniques were explored on the dataset. Multiple Regression (MR), Decision Trees (DT), Random Forests (RF), Neural Networks (NN), and Support Vector Machines (SVM). As per the paper, Multiple regression is widely used but can only learn linear mappings. Decision Trees and Random forests follow tree structure which are governed by IF-THEN rules. NNs are inspired by the human brain where multilayer perceptron's and logistic activation functions result in one output node with a linear function. SVM is used in conjunction with Radial Basis Function (RBF) which presents less hyperparameters and numerical difficulties than other kernels. Their choice for performance metrics were Root Mean Squared Error (RMSE) and Mean Absolute Deviation (MAD). Their best regressor was based on SVM with four meteorological inputs- Temperature, Relative Humidity, Rain and Wind and could predict forest fires with small forest fires which are also more frequent (Paulo Cortez, n.d.). As per their experiments, it is seen that 4 features out of 13, their SVM had the best performance. But feature selection does not always guarantee better performance. It may reduce dimensions of the data but may increase the possibility of false positives (Borja Seijo-Pardo, n.d.).Also, Datasets with larger dimensions might not always benefit from feature

selection. Since the paper mentions that multiple regression can't model nonlinear data accurately, we decided to build our model based on 3 algorithms- SVM, Decision Trees and Random Forests.

The paper "Pattern clustering of forest fires based on meteorological variables and its classification using hybrid data mining methods", published by Yong Poh Yu et al. explored 2 data mining techniques on Forest fires dataset from Montesinho Natural Park, Portugal. Clustering was used for pre-processing of the dataset. The historical meteorological variables were clustered using self-organizing map (SOM). Clustering was used because it can be trained without any supervision. SOM clusters inputs with similar characteristics into the same category. The reduction in dimensionality through clustering helps in improving performance in the later techniques. This clustered data was then used as input for 2 approaches- Back propagated neural network and Rule generation approach. Back-propagated NN is used to train feed-forward multilayer perceptron. The weight of each neuron is updated based on a activation function, which in their case, was sigmoid activation function. The NN took the classifier approach and classified the predictions into small, medium and large burned area. Rule base system followed a set of IF-THEN rules. As per their analysis, Back Propagated Neural network performed better than Rule based technique (Yong Poh Yu, n.d.). Their approach involved solving the problem using the classification technique. Therefore, it is unable to predict the burned area of the forest fire. It will only predict the magnitude of the forest fire. A regression approach will predict the rough estimate of burned area and its RMSE will give its error threshold.

The paper "Forest Fire Area Estimation using Support Vector Machine as an Approximator" explores the montesinho forest fire dataset with the Support Vector Machine approach. SVM with 2 kernel tricks – Radial Basis Function and Polynomial Function was used. The model performance was judged by 2 metrics- RMSE and MAE. (Nittaya Kerdprasop, n.d.) As per their findings, SVM with polynomial kernel gave the best performance- RMSE = 7.65 and MAE = 6.48. Compared to the results published by Paulo Cortez, RMSE is considerably lower. This was achieved without any feature selection. The error is considerably low, but the model training and testing times have not been published. Accurate predictions are necessary, but if the training and testing times are too long, then the models have no practical use.

METHODS

PRE-PROCESSING TECHNIQUES

ORDINAL ENCODING

The categorical data given in the dataset need to be converted into numerical data for the machine to understand. Hence, we have converted the months and days given in a categorical format to numerical format using LabelEncoder(). LabelEncoder() encodes the first unique value in data as 0, the second unique value as 1, and so on. The data encoding or our data is as follows:

Encoding for months:

Months	Encoded value
January	4
February	3
March	7
April	0
May	8
June	6
July	5
August	1
September	11
October	10
November	9
December	2

Encoding for days:

Days	Encoded Value
Monday	**1**
Tuesday	**5**
Wednesday	**6**
Thursday	**4**
Friday	**0**
Saturday	**2**
Sunday	**3**

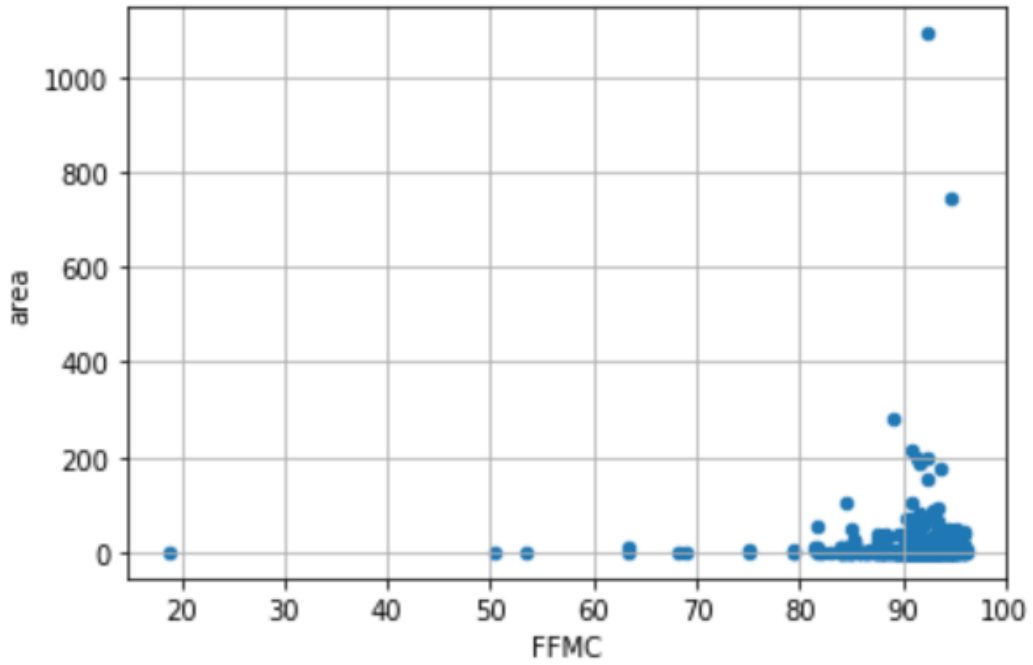
STANDARDIZATION OF DATA

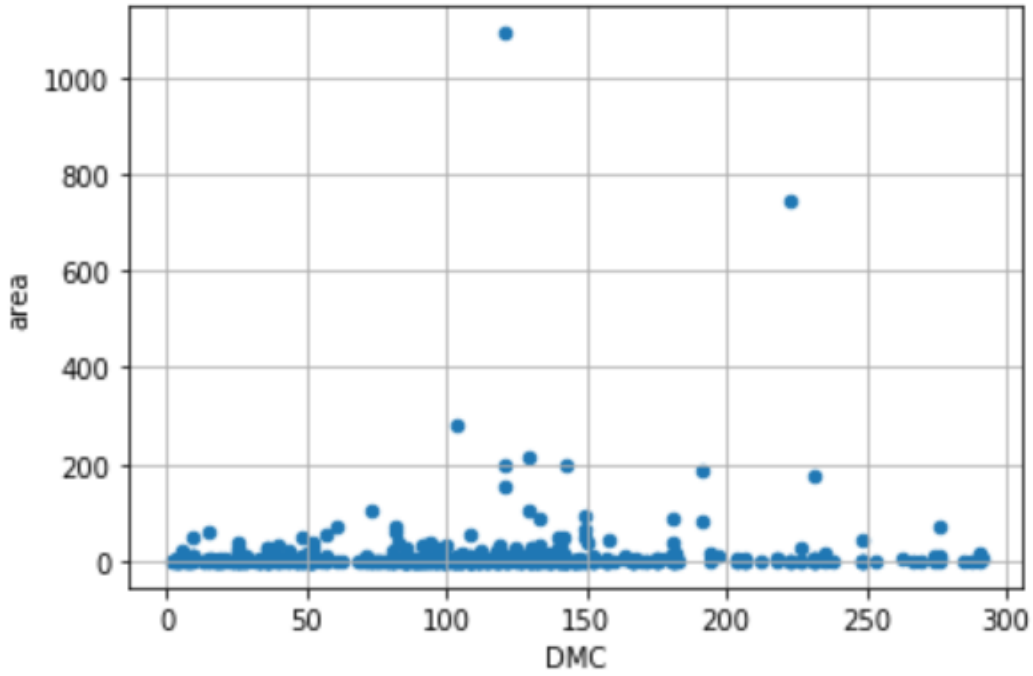
In order to reduce the variance in the magnitude of different features in the dataset, feature scaling needs to be done. This also helps in ensuring that each feature has an equal influence on the result obtained. We have used StandardScaler() for scaling the features of the given data.

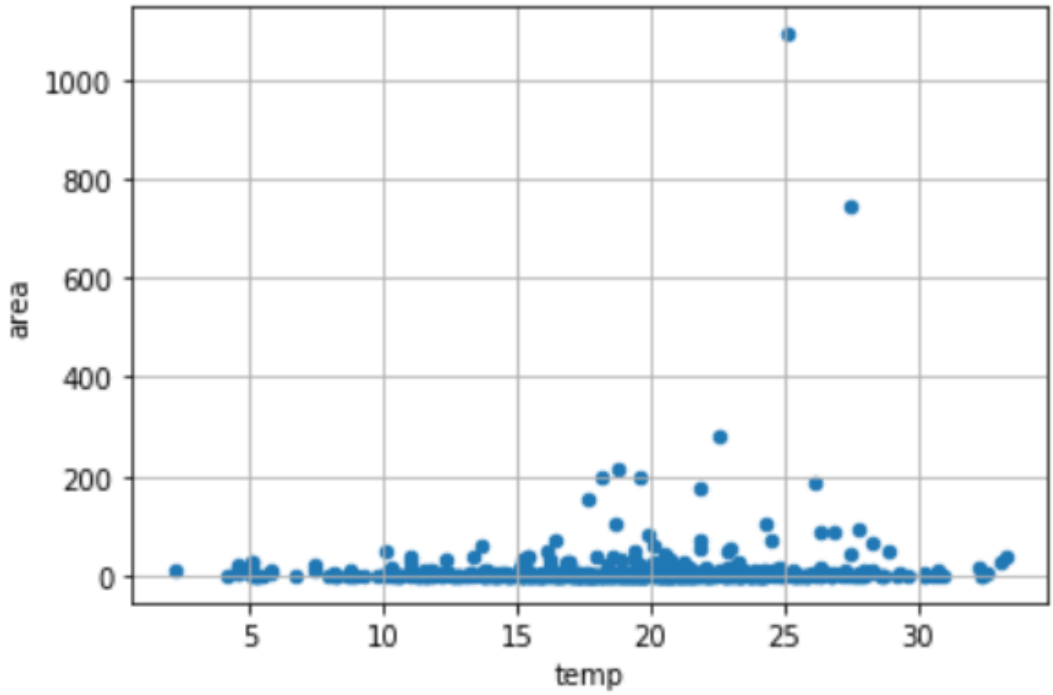
FEATURE SELECTION

In machine learning, selecting the features used for prediction is a vital task as it directly influences the accuracy and time taken by the model. The irrelevant or partially relevant data present can negatively impact the predicted result. In our model, we have used the top 3 features (temp, ffmc, dmc) which highly correlated with the area.

Scatter Plot with different 3 features are given below.







ALGORITHMS

SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a non-parametric technique which belongs to the supervised learning category. It has 4 main concepts:

Kernel: It is the function that maps lower dimensional data to higher dimension.

Hyper plane: In the case of SVM, it is the line that separates the data classes. However, in SVR, it is the line that will help us in predicting the continuous values or target value.

Boundary Line: It is the line that is at epsilon distance from the hyperplane.

Support Vectors: The data points closest to the boundary or on it are called support vectors. The distance should be minimum or least.

Support Vector Regressor (SVR) is based off of Support Vector Machine (SVM) with a few tweaks. SVM is used for classification where discrete values are predicted. SVR is used for regression problems where continuous values are used. SVR follows the equation below:

$$\min_{f} \|f\|_K^2 + C\sum_{i=1}^{l} |y_i - f(\mathbf{x}_i)|_\varepsilon$$

For linear inseparable datasets, soft margin and Kernel functions are used. Soft margin function determines the line that separates the data while tolerating a few misclassified points. Kernel trick utilizes existing features and creates new features by transforming existing ones. It works by generating new features by measuring the distance between all other data points to centers. Kernel function can be Linear, Radial basis and Polynomial. In our case, Radial basis function kernel was used. This pairing

of soft margin function and Kernel trick makes SVM a powerful machine learning technique and gave the best results on the forest fires dataset (Paulo Cortez, n.d.).

We implemented SVR using the support machine regressor provided by the SkLearn library. The hyperparameters that we tweaked to obtain the best results were Epsilon, Kernel and C (Penalty Parameter). Kernel was fixed to Radial Basis Function. The regressor was tested with the following Epsilon values-10, 0.1, 0.01, 0.001 and 0.0001 and Penalty values- 1, 0.1, 0.01 and 10. GridSearchCV was used along with make_pipeline to test these hyperparameters to find the best ones.

DECISION TREE

Decision Tree is a non-parametric supervised learning technique. It is made up of root nodes, internal nodes and leaf nodes. The attribute of the dataset is tested on the internal node, its branches represent the outcome and the leaf nodes show the class label (Decision taken). The classification rules are represented by the path from the root to leaf node. (Himani Sharma, n.d.)

- **Splitting:** The dataset is partitioned into subsets.
- **Decision Node:** The node that splits into further sub-nodes.
- **Leaf Node:** Nodes that represent the class of the datapoint.
- **Pruning:** Removal of sub-Nodes of a decision node is called Pruning.

In Regression trees, the regression model is fitted to each node to give the predicted values. (Loh, n.d.). Since each node is considered, the analysis of the decision can easily be done by traversing the branches. Therefore, we decided to use decision trees for our regression problem.

In the Sklearn DecisionTreeRegressor, the splitting is done based on mean squared error criterion. Maximum depth of the tree in our case lied between the range of 1 to 21. Minimum sample leaf values were 1,5,10,20,50 and 100. Best hyperparameters obtained by Grid search cross validation (GridSearchCV) method were 2 for Maximum depth and 1 for Minimum sample leaf values. The parameters were passed to GridSearchCV using make_pipeline function of SkLearn.

RANDOM FOREST

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. (Breiman, n.d.). Random forest algorithm belongs to the category of ensemble learning- Techniques which utilize multiple learning algorithms for prediction. Random forests overcome the drawback of decision trees- overfitting.

The Random Forest Algorithm works in 2 stages – Create random forests and Perform prediction on the generated forests.

Creation of random forests is done by first selecting a few features from the total number of features. Then the next node is calculated on the basis of best split point. The node is split into its children nodes. This process is repeated until max number of defined leaf nodes is reached. This is process is repeated a number of times to create a forest of trees.

The rules of randomly created decision trees are applied on the test features to predict the outcome. This prediction is stored as Target. Different random forests will have different precisions for the same test feature. Hence target votes are calculated and the prediction with the highest votes is the final prediction.

We implemented the RandomForestRegressor from the SkLearn. Ensemble Library. The following hyperparameters with their ranges were used for optimizing our model's performance- Max depth (5,10,15,20,50), Max leaf nodes (2,5,10), Minimum sample leaves (2,5,10) and Minimum sample split (2,5,10). To find out the best values for our hyperparameters, we passed the values to GridSearchCV function using the make_pipeline function.

Random forests are considered improved forms of decision trees because they are a combination of decision trees. This ensemble of decision trees makes them more robust. They work by dividing data into subsets which is preferred for datasets which have many dimensions.

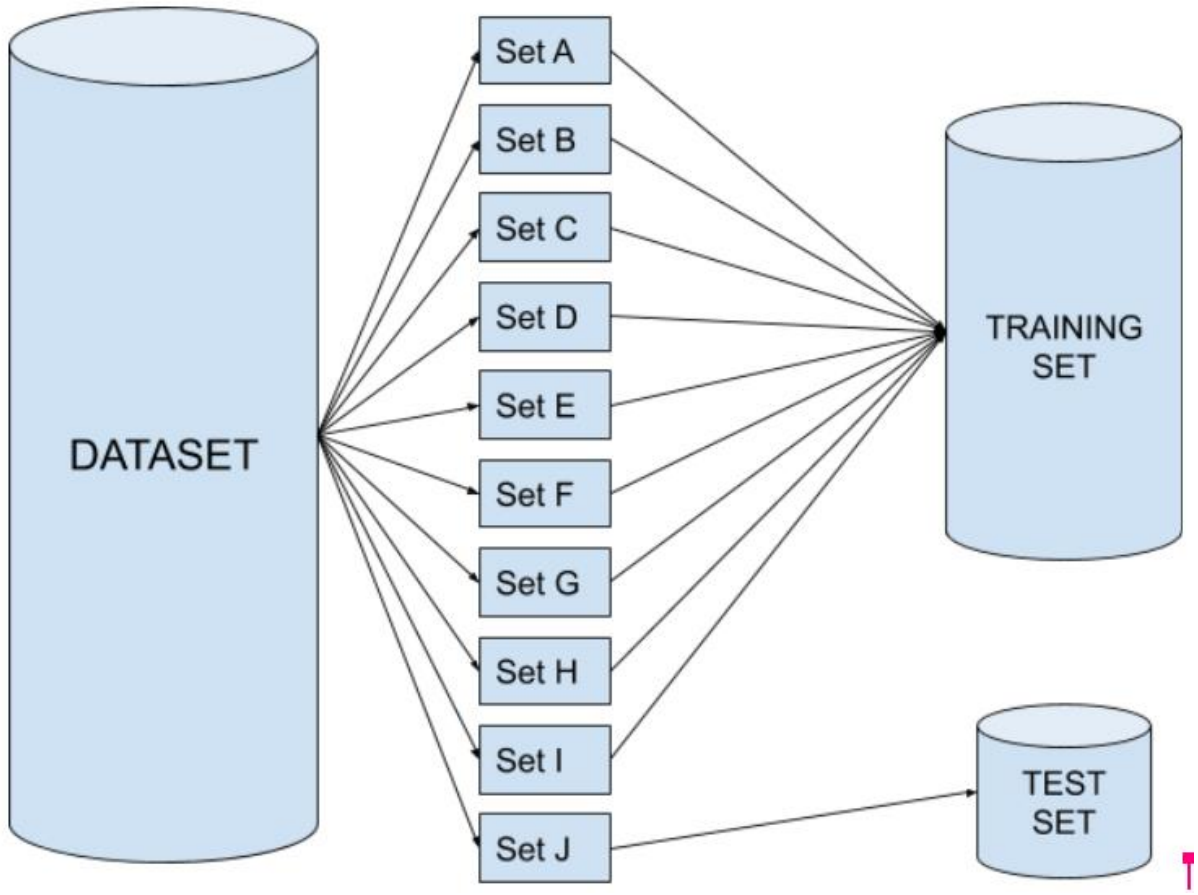
VALIDATION METHODS

CROSS VALIDATION

Once the training model is made, the efficiency of it needs to be tested to ensure that the model is giving accuracy as desired. Normally train-test split approach is used for analysing the performance of the model in which the given dataset is split into 80:20 or 70:30 ratio for training and validating the model respectively. However, this technique comes with a risk of high variance in the result. In other words, the accuracy obtained in one test sample can be very different from the one obtained in the different test sets. Therefore, we have used K-fold cross-validation, where k is the number of folds.

K-FOLD CROSS VALIDATION

In K-fold cross-validation, the issue of high variance in the model is handled by randomly choosing the training and testing samples. The given dataset is divided into K random sets of data and out of which K-1 sets are used for training and the remaining set is used for testing the model. Similarly, the process continues until each set or fold is used at least once for training and testing the data. In the end, the variance is calculated by averaging the results of entire folds defined.



For instance, in our model, we have used a 10-fold cross-validation technique. We divide the forest fire dataset into 10 sets (say, Set A, Set B, Set C, Set D ……. Set J), wherein first fold Set A to Set I are

used for training and Set J is used for testing. In the second fold, Set A to Set H and Set J is considered for training whereas Set I is used for testing data. Thereby ensuring every data sample gets an equal chance of appearing in both the training and testing samples.

GRID SEARCH CROSS VALIDATION

While building machine learning models we usually have two types of parameters, one, which the model learns while getting trained and the other which we pass to the machine learning model. Normally the hyperparameters are passed randomly and the ones giving the best results are chosen. However, this approach can be exhaustive. Therefore, we have used the grid search cross-validation, for selecting the best hyperparameters. Grid Search algorithm automatically finds the values of hyperparameters from the given set of multiples for which the model gives the best results.

EXPERIMENTS AND DISCUSSION

We automated the process of tweaking the hyperparameters by using grid search CV.

EXPERIMENTS WITH DIFFERENT TECHNIQUES

SUPPORT VECTOR REGRESSOR

Hyperparameters values tested:

- Epsilon values: 10, 0.1, 0.01, 0.001 and 0.0001
- Penalty values: 1, 0.1, 0.01 and 10

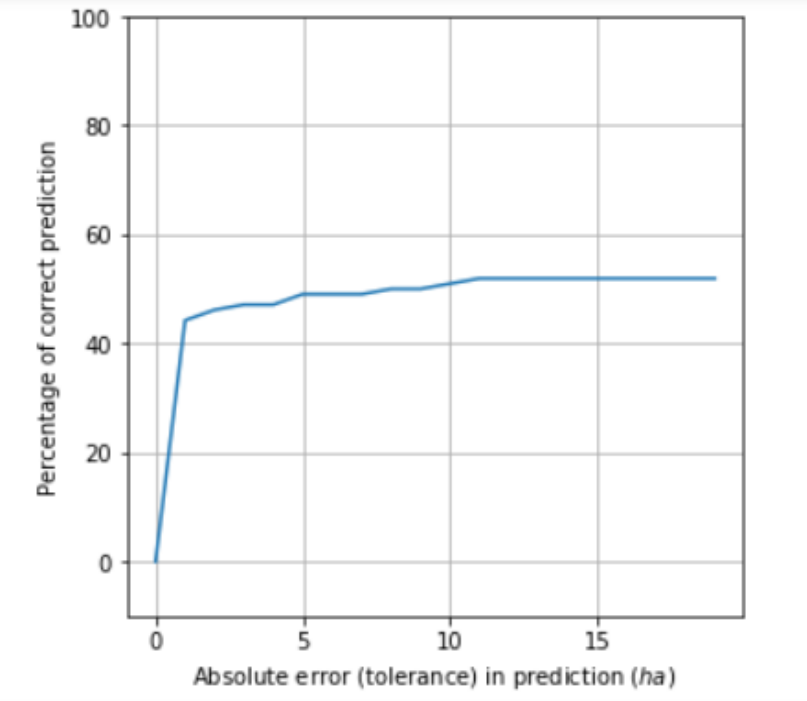
C	0.01	1	0.1
Epsilion	0.1	0.01	0.001
RMSE	26.72	26.73	26.75

Best parameters found by Grid Search CV:

- **C: 0.01**
- **epsilon: 0.1**
- **kernel: 'rbf'**

RMSE: 26.72
MAD: 2.77

REC CURVE for Support Vector Regressor



DECISION TREE REGRESSOR

Hyperparameters:

- Maximum depth: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21.
- Minimum sample leaf values: 1, 5, 10, 20, 50 and 100.

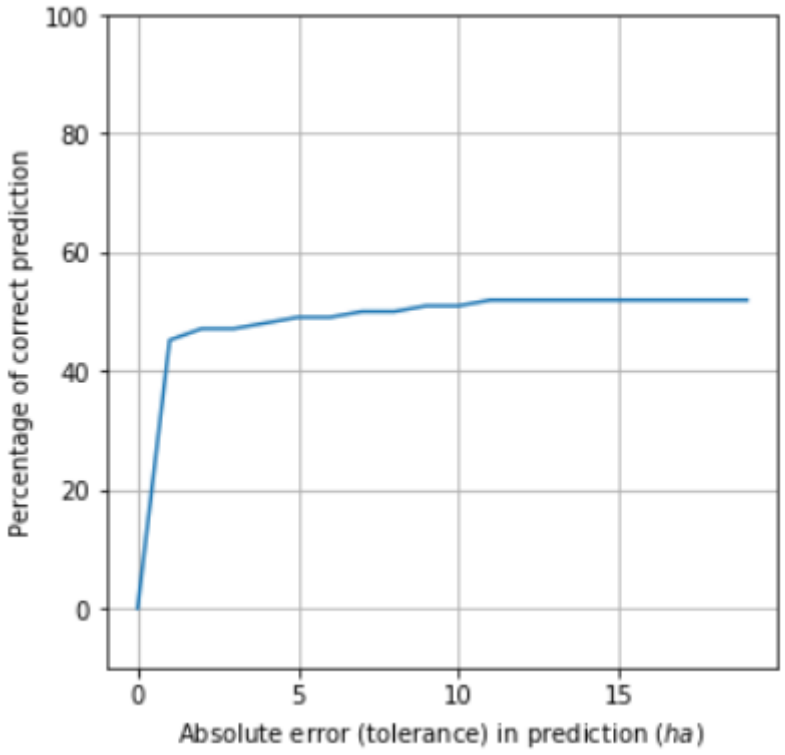
Max Depth	2	1	2
Min Sample Leaf	1	100	50
RMSE	26.75	26.87	26.89

Best parameters found by Grid Search:

- **decisiontreeregressor_max_depth: 2**
- **decisiontreeregressor_min_samples_leaf: 1**

RMSE: 26.75
MAD: 0.0

REC CURVE for Decision Tree Regressor



RANDOM FOREST REGRESSOR

Hyperparameters:

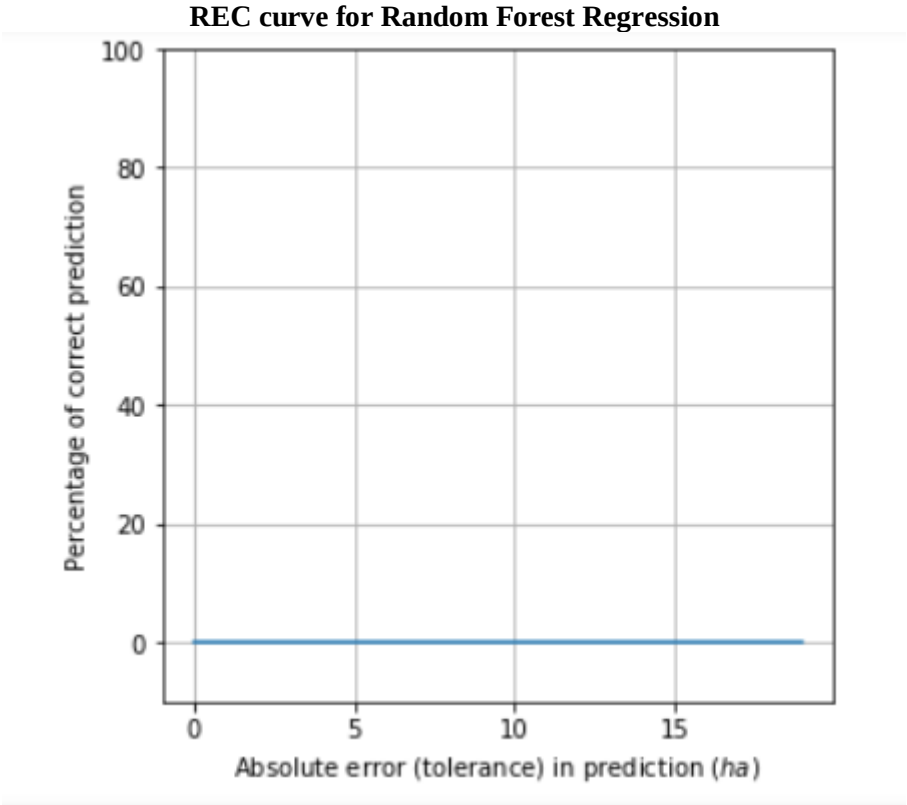
- Max depth: 5,10,15,20,50
- Max leaf nodes: 2,5,10
- Minimum sample leaves: 2,5,10
- Minimum sample split: 2,5,10

Max Depth	5	15	20
Max Leaf Nodes	2	10	5
Min Sample Leafs	10	5	2
Min Sample Split	5	2	10
RMSE	26.97	34.57	35.34

Best parameters found by Grid Search:

- **max_depth: 20**
- **max_leaf_nodes: 2**
- **min_samples_leaf: 10**
- **min_samples_split: 5**

RMSE: 26.97
MAD: 3.81



DISCUSSION OF RESULTS

Regression Model	RMSE	MAD
Support Vector Regressor	26.72	2.77
Decision Tree Regressor	26.75	0.0
Random Forest Regressor	26.97	3.81

Support Vector Regression			
C	0.01	1	0.1
Epsilon	0.1	0.01	0.001
RMSE	26.72	26.73	26.75

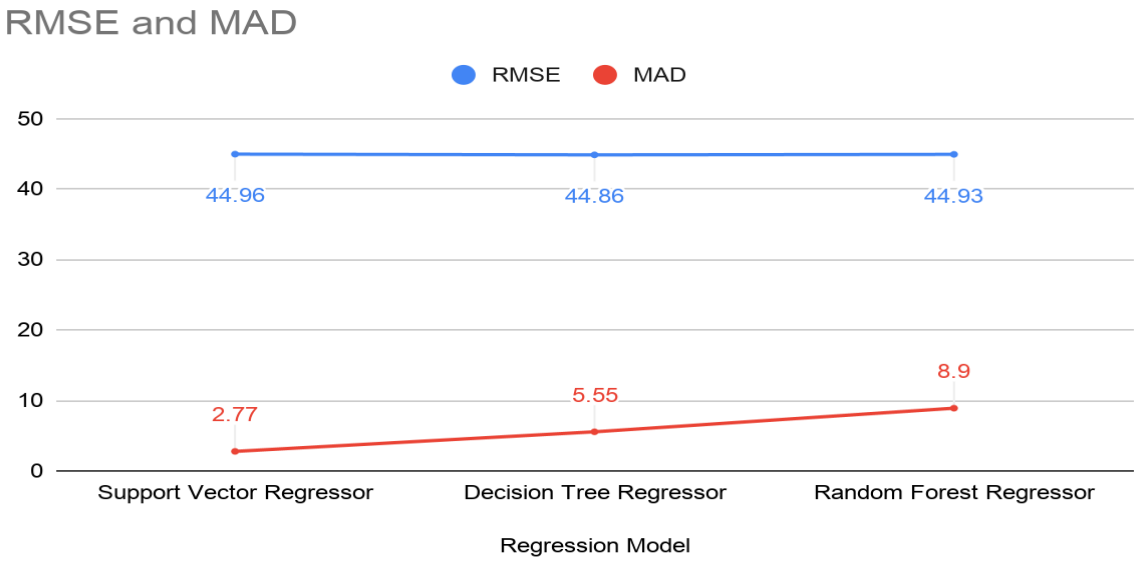
Decision Tree Regression			
Max Depth	2	1	2
Min Sample Leaf	1	100	50
RMSE	26.75	26.87	26.89

Random Forest Regression			
Max Depth	5	15	20
Max Leaf Nodes	2	10	5
Min Sample Leaf	10	5	2
Min Sample Split	5	2	10
RMSE	26.97	34.57	35.34

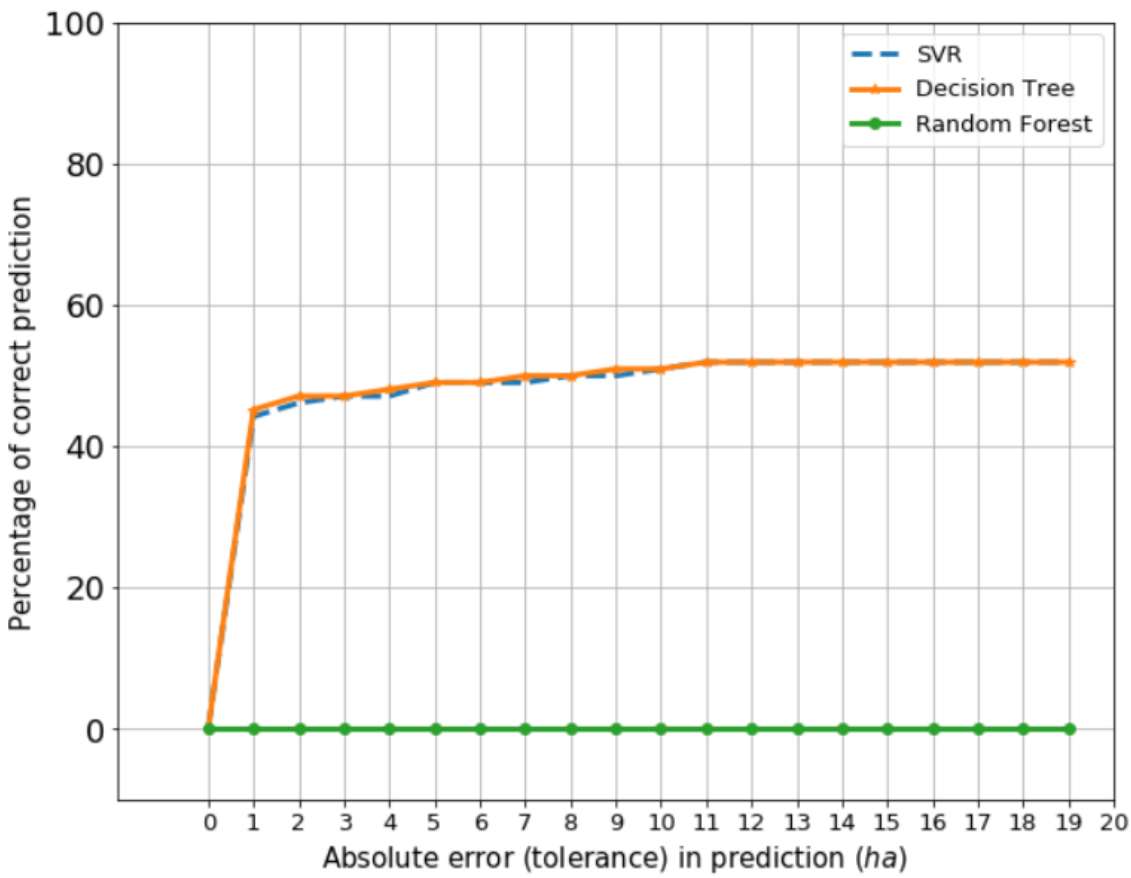
The different values for the hyperparameters were decided through literature review and educated guess. As per the tables above tweaking the hyperparameters gave us marginal improvements for SVR and Decision Trees. For Decision Trees, change in hyperparameters does not result in greater change in RMSE.

In the case of Random Forest, tweaking of hyperparameters showed the most improvement. During our experiments with Random Forest, we observed that, RMSE increased with max depth. More the depth, greater the RMSE.

In case of SVR, we used different C (penalty) and Epsilon values to get our best hyperparameters using gridSearchCV method. There was no considerable difference between results for different parameters although choosing C value as 0.01 and Epsilon value as 0.1 gave us the minimum RMSE of 26.72



REC Curve for all models



PERSONAL REFLECTION

The forest fire dataset was an interesting dataset to work on. We got to implement different machine learning techniques to solve the problem. SVM, Decision Trees and Random Forests are usually used as classifiers, so implementing the algorithms as regressors was challenging. Our initial assumptions were that using libraries will make the assignment easier. The assignment presented its own set of challenges. The dataset size was comparatively small, so proper division of the data into testing and training sets was crucial for our model to work efficiently. Feature selection was also not as straightforward as we initially thought. Recursive feature elimination technique doesn't consider the hidden correlation between features. As per our experiments, we concluded that SVR is our best model and it has RMSE value lower than the benchmark mentioned.

CONCLUSION AND FUTURE WORK

It is evident from the research of Hancock that forest fires devastating impact on human lives as well as properties (Hancock, n.d.). Moreover, if occurred it takes lots of time and effort to tackle with the damage caused and make the situation normal for public. However, this adverse effect of forest fires can be prevented if predicted at right time. A lot of research has been done on predicting forest fire using various methods such as Neural Networks, Support Vector Regressor, Decision Trees and Random Forests. However, Support Vector Regressor used in the research we referred either has more error or lacks vital parameters for accuracy such as runtime, which also plays a vital role in evaluating a model. In addition to this, Neural Network used in the research was unable to predict the accurate area (dimensions) of forest fire as they used classification technique. We therefore, built our model using regression techniques such as Support Vector Regressor, Decision Trees and Random Forest in which Support Vector Regressor outperformed others in terms of RMSE and MAD. Our model performed better than previously researched Support Vector regressor because of the use of GridSearch, which automatically tuned the hyperparameters for us.

Currently, we have considered only the off-line data provided to us which contains montesinho natural park forest fires data. In the future, we hope to work with real-time data collected using IoT devices. Nowadays, IoT sensors help in monitoring various environmental factors such as carbon dioxide, carbon monoxide and other polluting gases present in the atmosphere. Using this data, we can further predict the features which are responsible for adverse weather conditions that lead to natural calamities such as forest fires. Also, big data technologies will be used to collect data in real-time will be huge. Integrating IoT with machine learning and big data will help us in continuously monitoring the atmospheric condition as well as in the early detection of forest fires. Early detection will give us enough time to carry out security procedures. Hence, the need for mobilization of different emergency agencies and services can be fulfilled, thereby avoiding the risk of damage and loss of lives.

APPENDIX

1) Download the data set from below link
   https://archive.ics.uci.edu/ml/machine-learning-databases/forest-fires/

   Data Set Name: forestfires.csv
   Please make sure the file name is correct

2) Place the data set in the same folder as code files

CODE FILE 1: group5_best_algorithm1.ipynb

This file contains out best regressor: Support Vector Regressor.

- Restart and Run all

CODE FILE 2: group5_other_algorithms.ipynb

This file contains the other two regressors: Random Forest Regressor and Decision Tree Regressor

- Restart and Run all
- You will be given a choice for the algorithm, 1 is for Decision Tree Regressor and 2 is for Random Forest Regressor
- Please enter your choice and the corresponding results will be displayed.

Bibliography

1. Borja Seijo-Pardo, A. A.-B., n.d. *Biases in feature selection with missing data.* [Online] Available at: https://www.sciencedirect.com/science/article/pii/S0925231219301493

2. Breiman, L., n.d. *Random Forests.* [Online] Available at: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

3. Daniela Stojanova, P. P. A. K. S. D. K. T., n.d. *LEARNING TO PREDICT FOREST FIRES WITH DIFFERENT DATA MINING TECHNIQUES.* [Online] Available at: https://s3.amazonaws.com/academia.edu.documents/30570649/10.1.1.116.2555.pdf?response-content-disposition=inline%3B%20filename%3DLEARNING_TO_PREDICT_FOREST_FIRES_WITH_DI.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20191107

4. Hancock, L., n.d. *Forest Fires: the good and the bad..* [Online] Available at: https://www.worldwildlife.org/stories/forest-fires-the-good-and-the-bad

5. Himani Sharma, S. K., n.d. *A Survey on Decision Tree Algorithms of Classification in Data Mining.* [Online] Available at: https://www.researchgate.net/publication/324941161_A_Survey_on_Decision_Tree_Algorithms_of_Classification_in_Data_Mining

6. Loh, W.-Y., n.d. *Classification and Regression Trees.* [Online] Available at: https://www.researchgate.net/publication/227658748_Classification_and_Regression_Trees

7. Nittaya Kerdprasop, P. P. P. C. K. K., n.d. *Forest Fire Area Estimation using Support Vector Machine as an Approximator.* [Online] Available at: https://pdfs.semanticscholar.org/afb3/be2204f663f69f9fd9a1181daa47dc343165.pdf

8. Paulo Cortez, A. M., n.d. *A data mining approach to predict forest fires using meteorological data,.* [Online] Available at: http://www3.dsi.uminho.pt/pcortez/fires.pdf

9. Yong Poh Yu, R. O. R. D. H. M. K. S. A. R. N., n.d. *Pattern Clustering of forest fires based on meteorological variables and its classification using hybrid data mining methods.* [Online] Available at: https://pdfs.semanticscholar.org/98b6/dcfcdb485ddadba6f17337b4f3702cb5e8d9.pdf